

Practical 4

Aim:

To Implement Single-row functions.

(1) Write a query to display the current date. Label the column Date

Query: SELECT SYSDATE as "DATE" FROM dual

Output:

User: 22DCE006

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select SYSDATE as "DATE" from DUAL
```

Results Explain Describe Saved SQL History

DATE
05-FEB-24

1 rows returned in 0.00 seconds [CSV Export](#)

(2) For each employee, display the employee number, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary

Query: select emp_name,emp_no,emp_sal,(emp_sal + emp_sal*0.15) AS NEW_SAL from employee

Output:

User: 22DCE006

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select emp_name,emp_no,emp_sal,(emp_sal + emp_sal*0.15) AS NEW_SAL from employee
```

Results Explain Describe Saved SQL History

EMP_NAME	EMP_NO	EMP_SAL	NEW_SAL
smith	101	800	920
snehal	102	1600	1840
David	103	1100	1265
aman	104	3000	3450
anita	105	5000	5750
sneha	106	2450	2817.5
anamika	107	2975	3421.25

7 rows returned in 0.02 seconds [CSV Export](#)

(3) Modify your query no (2) to add a column that subtracts the old salary from the new salary. Label the column Increase

Query: select emp_name,emp_no,emp_sal,(emp_sal + emp_sal*0.15) AS NEW_SAL,((emp_sal +emp_sal*0.15)-emp_sal) INCREASE_SAL from employee

Output:

User: 22DCE006
Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select emp_name,emp_no,emp_sal,(emp_sal + emp_sal*0.15) AS NEW_SAL,((emp_sal +emp_sal*0.15)-emp_sal) INCREASE_SAL from employee
```

Results Explain Describe Saved SQL History

EMP_NAME	EMP_NO	EMP_SAL	NEW_SAL	INCREASE_SAL
smith	101	800	920	120
snehal	102	1600	1840	240
David	103	1100	1265	165
aman	104	3000	3450	450
anita	105	5000	5750	750
sneha	106	2450	2817.5	367.5
anamika	107	2975	3421.25	446.25

7 rows returned in 0.02 seconds [CSV Export](#)

(4) Write a query that displays the employee's names with the first letter capitalized and all other letters lowercase, and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names.

Query: select initcap(emp_name) AS EMPLOYEE_NAME,length(emp_name) as NAME_LENGTH from employee where emp_name like 'j%' or emp_name like 'a%' or emp_name like 'm%' order by l_name

Output:

User: 22DCE006
Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select initcap(emp_name) AS EMPLOYEE_NAME,length(emp_name) as NAME_LENGTH from employee where emp_name like 'j%' or emp_name like 'a%' or emp_name like 'm%' order by l_name
```

Results Explain Describe Saved SQL History

EMPLOYEE_NAME	NAME_LENGTH
Anamika	7
Anita	5
Aman	4

3 rows returned in 0.00 seconds [CSV Export](#)

(5) Write a query that produces the following for each employee:

<employee last name> earns <salary> monthly

Query: `select emp_name || ' earns ' || emp_sal as emp_name_sal from employee`

Output:

User: 22DCE006

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select emp_name || ' earns ' || emp_sal as emp_name_sal from employee
```

Results Explain Describe Saved SQL History

EMP_NAME_SAL
smith earns 800
snehal earns 1600
David earns 1100
aman earns 3000
anita earns 5000
sneha earns 2450
anamika earns 2975

7 rows returned in 0.00 seconds [CSV Export](#)

(6) Display the name, date, number of months employed and day of the week on which the employee has started. Order the results by the day of the week starting with Monday.

Query: `select emp_name,hiredate,round(months_between(sysdate,hiredate),0) as "MONTH_WORKED", to_char(hiredate,'DAY') AS "DAY_OF_WEEKS" from employee order by (hiredate-NEXT_DAY(hiredate,'monday'))`

Output:

User: 22DCE006

Home > SQL > SQL Commands

☒ Autocommit Display 10 Save

```
select emp_name,hiredate,round(months_between(sysdate,hiredate),0) as "MONTH_WORKED", to_char(hiredate,'DAY') AS "DAY_OF_WEEKS" FROM EMPLOYEE
order by (hiredate-NEXT_DAY(hiredate,'monday'))
```

Results Explain Describe Saved SQL History

EMP_NAME	HIREDATE	MONTH_WORKED	DAY_OF_WEEKS
anamika	15-JUL-97	319	TUESDAY
David	30-NOV-95	338	THURSDAY
snehal	14-MAR-96	335	THURSDAY
aman	02-OCT-97	316	THURSDAY
anita	01-JAN-98	313	THURSDAY
sneha	26-SEP-97	316	FRIDAY
smith	09-AUG-96	330	FRIDAY

7 rows returned in 0.00 seconds [CSV Export](#)

(7) Display the date of emp in a format that appears as Seventh of June 1994 12:00:00 AM.

Query: `select to_char(a_date,'DDth Month yyyy HH:MM:SS') from deposit`

Output:

User: 22DCE006

Home > SQL > **SQL Commands**

☒ Autocommit Display 10

```
select to_char(a_date,'DDth Month yyyy HH:MM:SS') from deposit
```

Results Explain Describe Saved SQL History

TO_CHAR(A_DATE,'DDTHMONTHYYYYHH:MM:SS')
01ST January 2006 12:01:00
15TH July 2006 12:07:00
12TH March 2006 12:03:00
17TH September 2006 12:09:00
19TH November 2006 12:11:00
21ST December 2006 12:12:00

6 rows returned in 0.00 seconds [CSV Export](#)

(8) Write a query to calculate the annual compensation of all employees (sal +comm.).

Query: `select sum(emp_sal +emp_comm) from employee`

Output:

User: 22DCE006

Home > SQL > **SQL Commands**

☒ Autocommit Display 10

```
select sum(emp_sal +emp_comm) from employee
```

Results Explain Describe Saved SQL History

SUM(EMP_SAL+EMP_COMM)
85450

1 rows returned in 0.01 seconds [CSV Export](#)

Conclusion: From this practical I learned about different SQL commands that can be used for data manipulation and single row SQL functions which are useful to getting one output per row.

Staff Signature:

Grade:

Remarks by the Staff: